The AI Revolution: LLM and AI agents

Yinghan Long
Dec 1, 2025

Outline

- LLM
  - **LLM Architecture**
- LLM Training
  - Pretraining
  - Supervised Finetuning
  - **Post training**
  - Reasoning
- AI Agent
  - Agent design basics
  - Deep Research

Best ways to learn AI

- Read classical and the latest influential research papers
- Write code with various open-source AI frameworks
- Use AI!

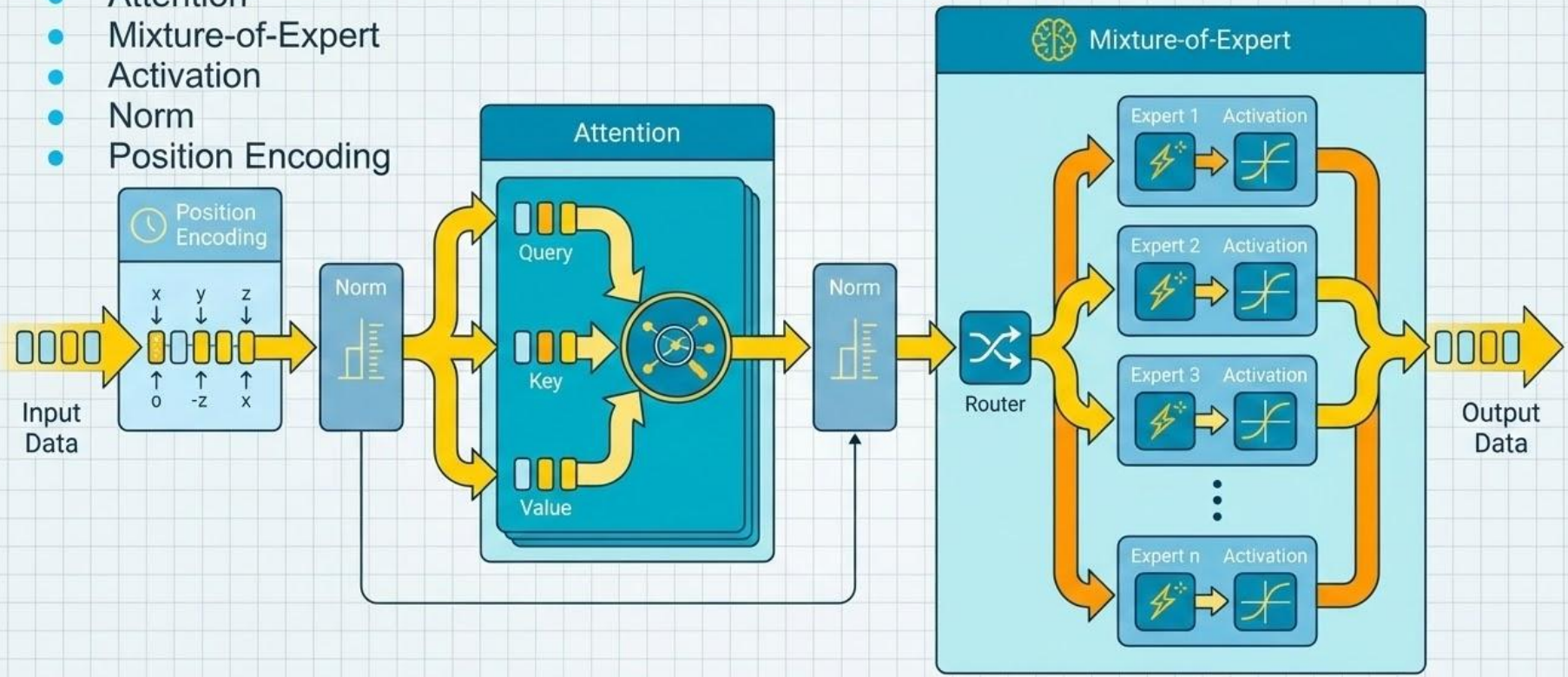
arXiv

GitHub

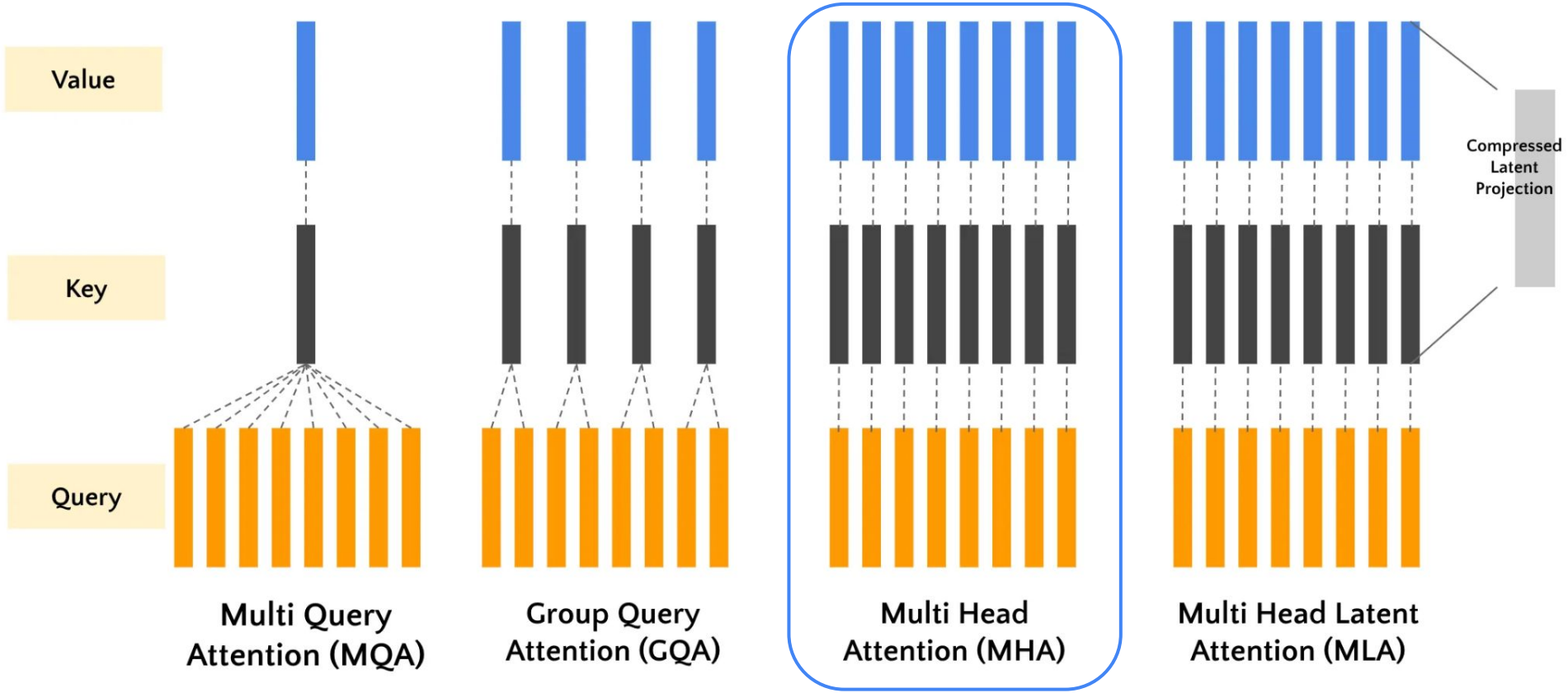
Hugging Face

LLM Components

- Attention
- Mixture-of-Expert
- Activation
- Norm
- Position Encoding

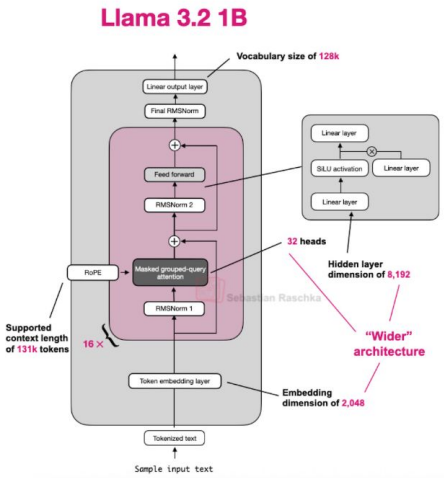


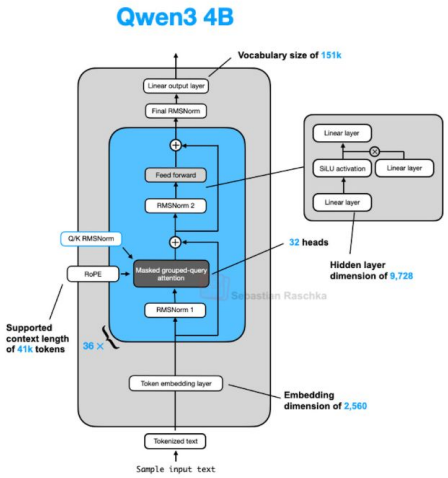
Attention – MQA | GQA | MHA | MLA

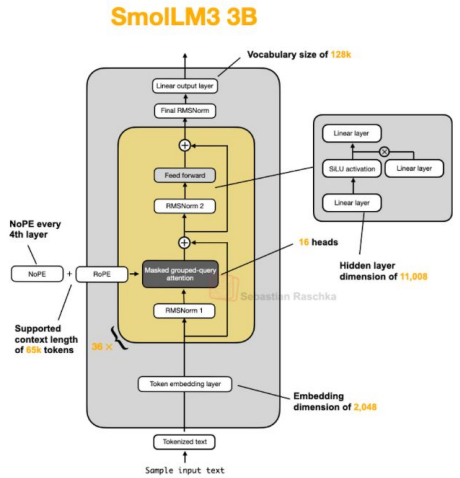


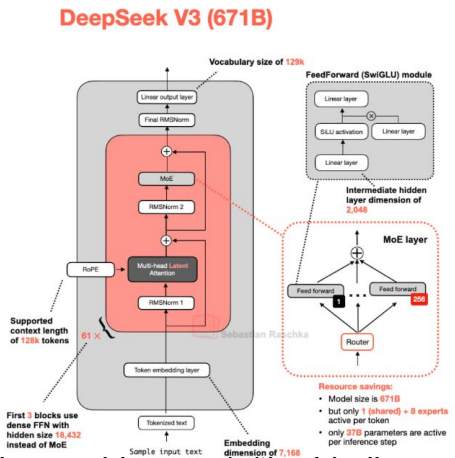
https://verticalserve.medium.com/group-query-attention-58283b337c65

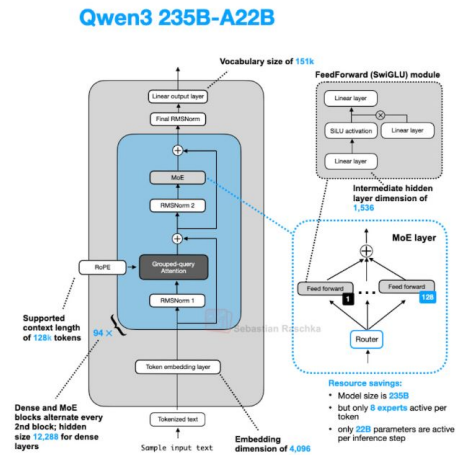
Open-source LLM

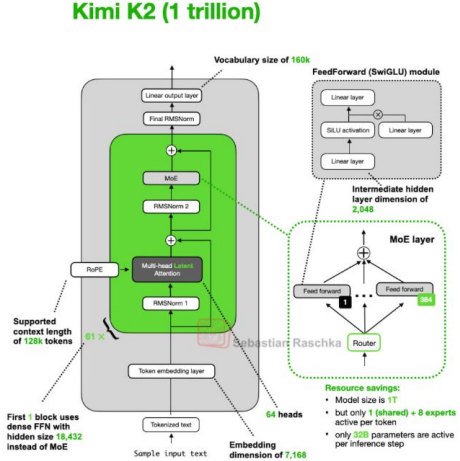








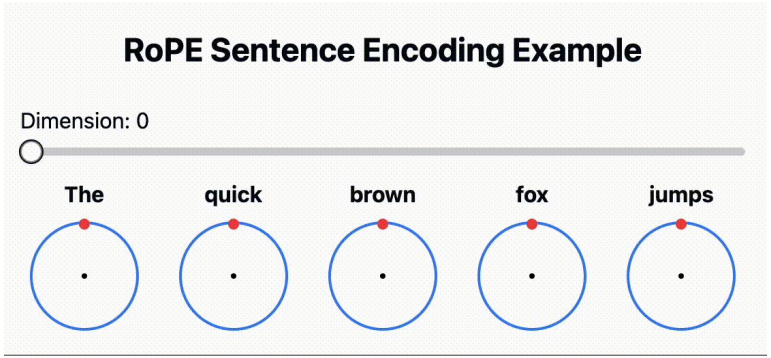




- https://magazine.sebastianraschka.com/p/the-big-llm-architecture-comparison
- https://github.com/rasbt/LLMs-from-scratch

Positional Encoding

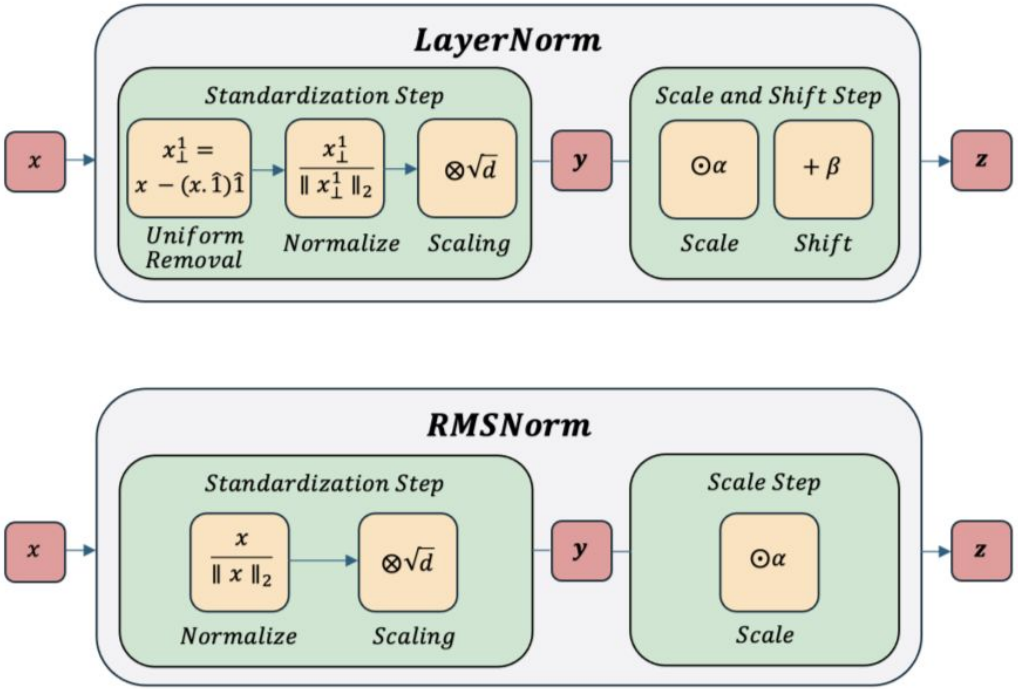
- Rotary Positional Encoding (RoPE)
- Encodes the absolute position with a rotation matrix and meanwhile incorporates the explicit relative position dependency



https://medium.com/@himankvjain/the-rope-effect-untangling-positional-encoding-in-ai-language-models-1bf0ab46776b

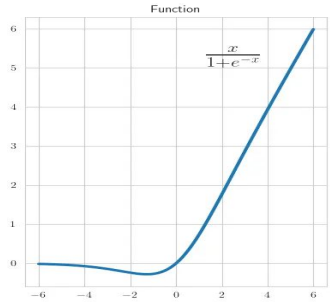
Normalization

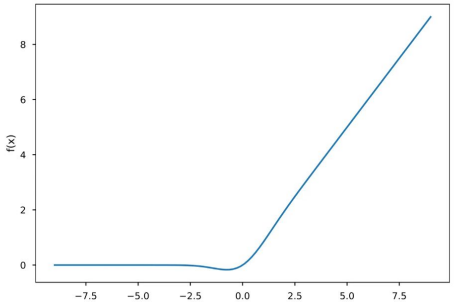
- LayerNorm (Earlier models such as GPT2/3)
- Root Mean Square Normalization (RMSNorm) (Llama and others)



Activation

$$\text{swish}(x) = x\,\text{sigmoid}(\beta x) = \frac{x}{1+e^{-\beta x}}$$





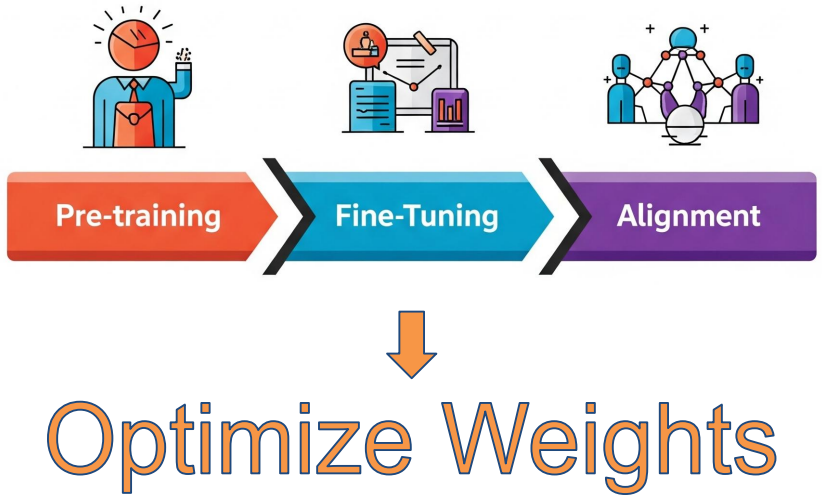
Feature	SwiGLU	GELU
Definition	$SwiGLU(x, W, V, b, c\beta) = Swish_\beta(xW + b) \otimes (xV + c)$	Gaussian error cumulative distribution function x*\phi (x)
Structure	A Gated Linear Unit (GLU) where the gating function is Swish (SiLU).	A non-gated activation function.
Mechanism	Combines an input with a second linear projection that has been passed through the Swish function. This gating allows the model to dynamically control which features are activated or deactivated.	Applies a single, smooth curve to all input dimensions.
Key Advantage	The gating mechanism provides more nuanced control over information flow, leading to better feature learning.	Simpler structure compared to gated activations.
Example Use	Standardized in models like PaLM and Llama.	Used in models like BERT

Three Stages of LLM Training

01 Pre-training: Unsupervised learning on massive datasets to predict the next token, learning the language itself.

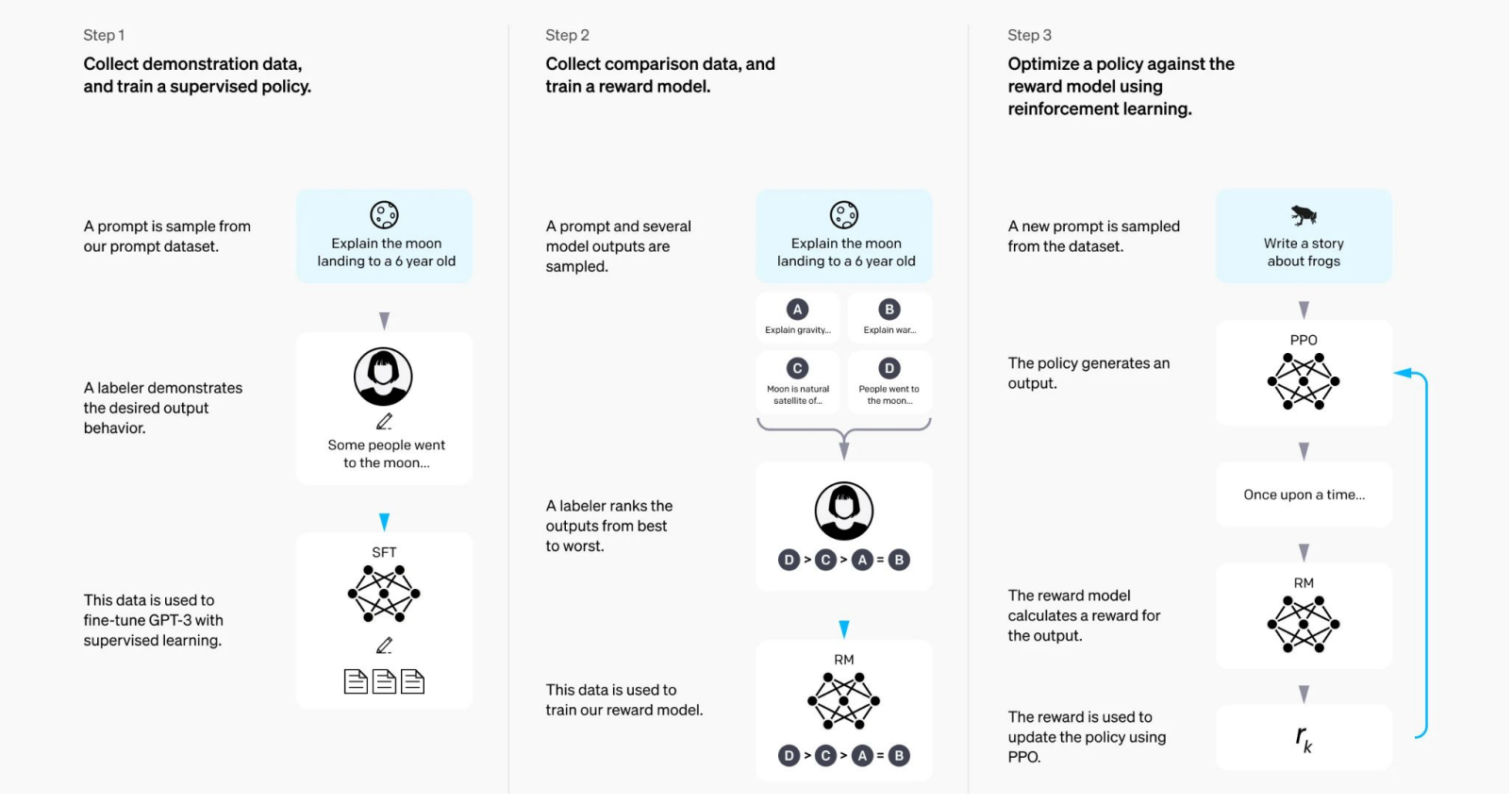
02 SFT (Supervised Fine-Tuning): Learning to follow instructions on curated human-written instruction-response pairs.

03 Post-Training (Alignment): Refining behavior to align model outputs with human preferences and safety via Reinforcement Learning.



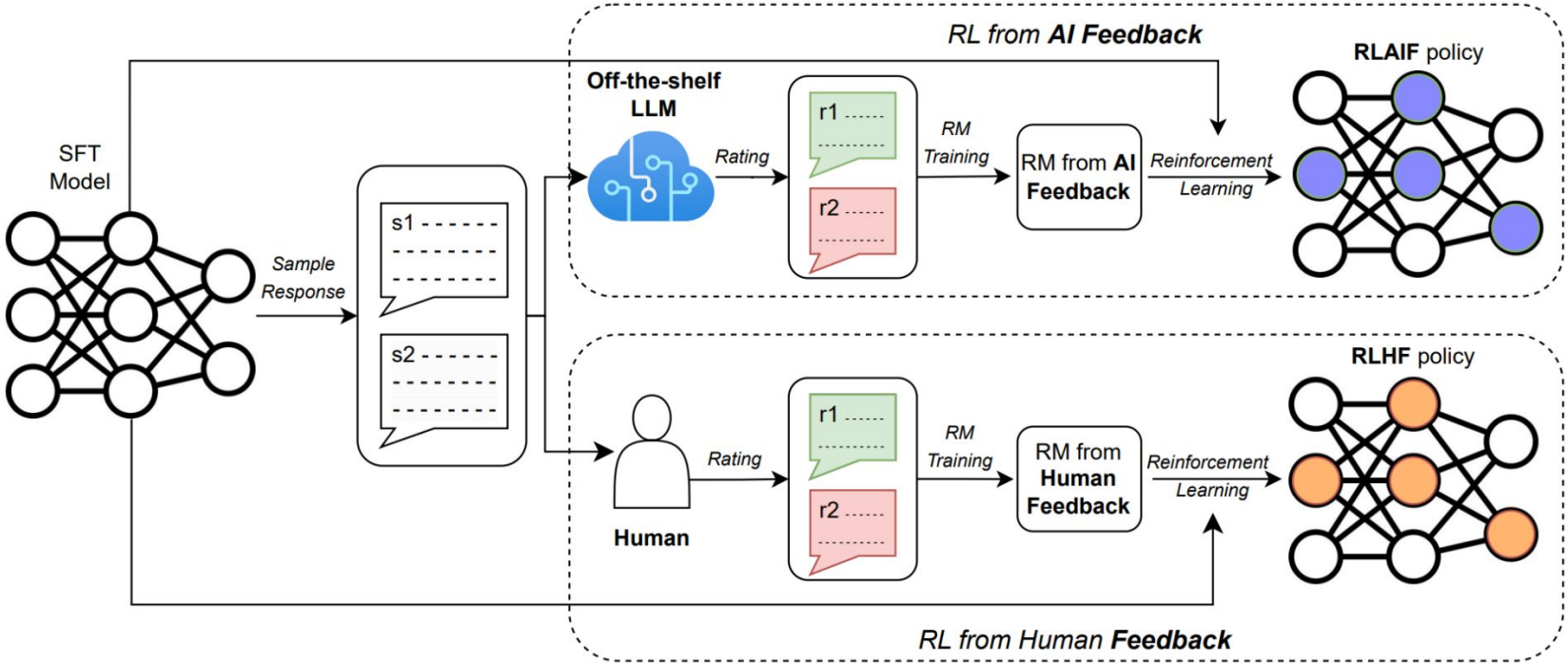
Post-training: Reinforcement Learning from Human Feedback

- RLHF (OpenAI, 2022)



RLAIF: Scaling Reinforcement Learning from Human Feedback with AI Feedback

- RLHF (Traditional): Requires feedback from human annotators
- RLAIF: AI as the judge



Reinforcement Learning with Verifiable Feedback (RLVR)

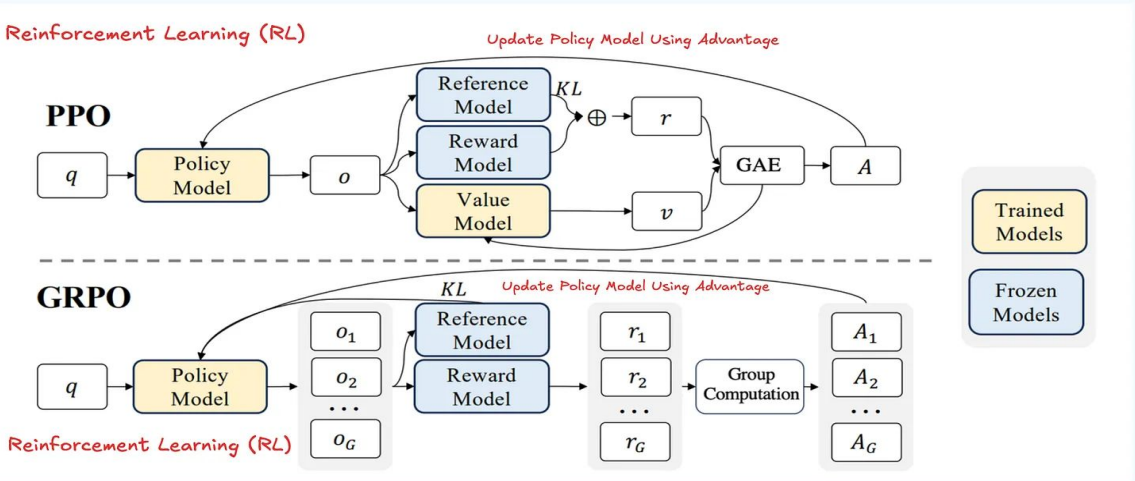
- **Automated Verification:** Instead of human evaluators, RLVF uses **predefined rules**, ground-truth answers, or code execution to check the correctness of an output. For example, a math problem's final answer can be numerically checked, or code can be run against test cases.
- **Objective Rewards:** The verifier provides a clear, objective reward signal, typically binary (correct/incorrect) or a soft probability score, which is used to update the model's policy via reinforcement learning algorithms.
- **Focus on Reasoning**: The approach incentivizes the model to develop robust, logical reasoning capabilities that lead to a provably correct answer, rather than "hacking" the system with superficial patterns or shortcuts.

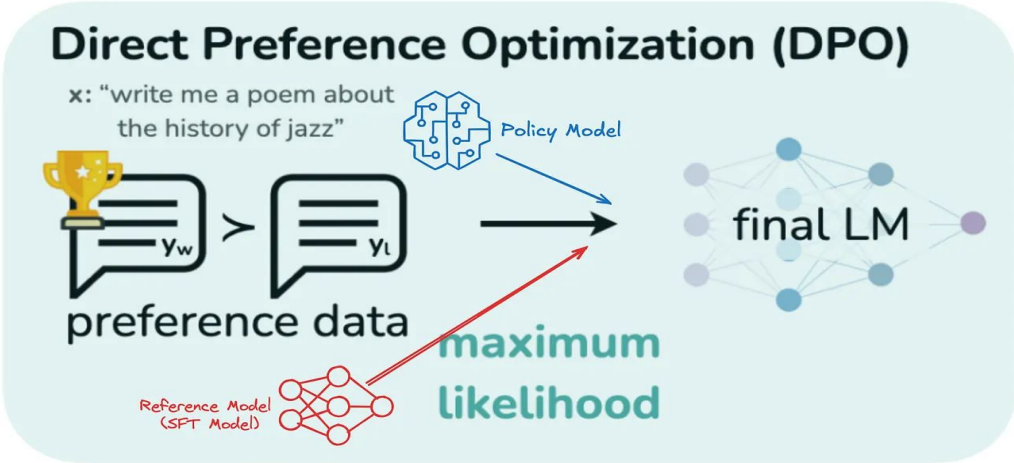
PPO vs DPO vs GPRO

PPO (GPT4): Proximal Policy Optimization; Requires a complex, separate Reward Model, often leading to unstable training with high complexity.

GPRO(Deepseek-R1): Group Relative Policy Optimization

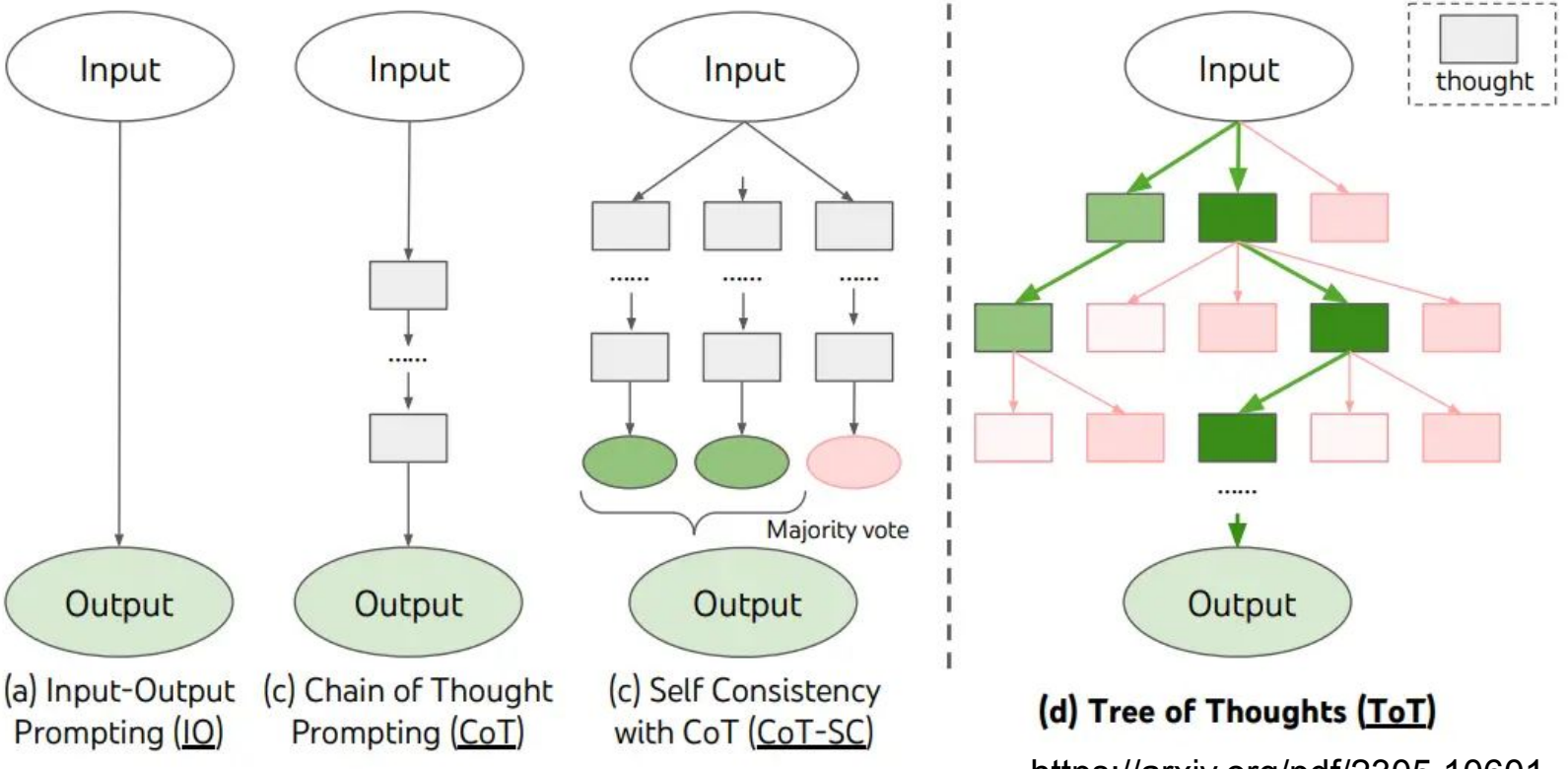
DPO (Llama3, Mistral): **No reward model!** Directly optimizes the policy using reference pairs, resulting in lower complexity and more stable training, converging like standard methods.





https://aiexpjourney.substack.com/p/the-best-way-to-understand-ppo-grpo

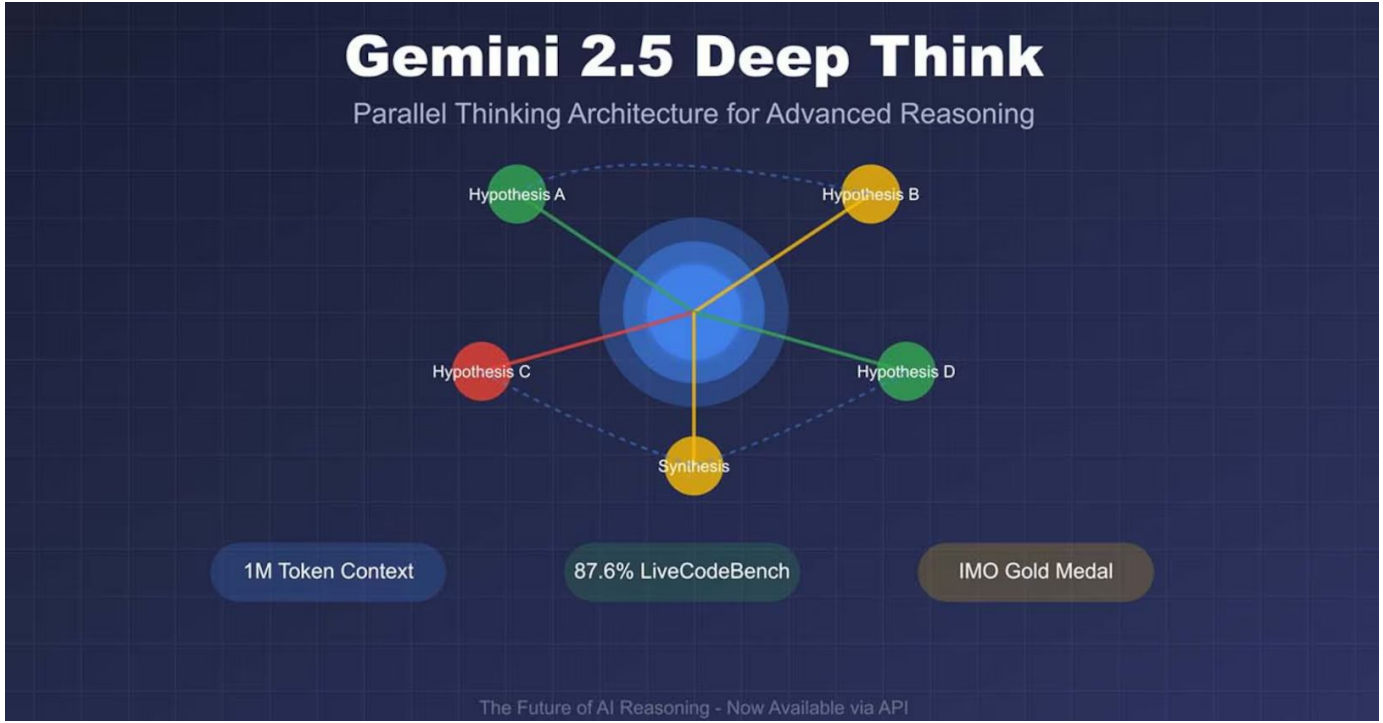
Reasoning Models: Another revolution



https://arxiv.org/pdf/2305.10601

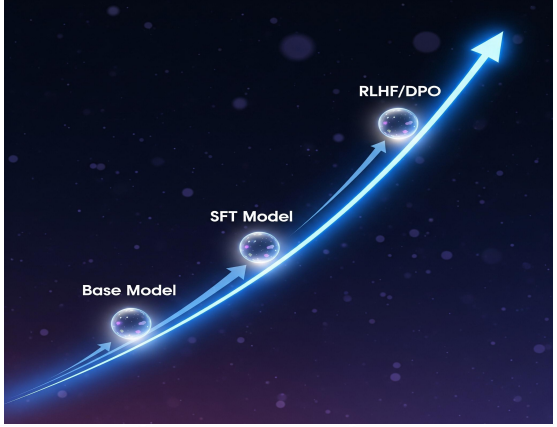
Reasoning Models: Another revolution

- Gemini Deep Think (Parallel)



LLM Evaluation

- MMLU
- More complex benchmark: Human's Last Exam
- LMArena: https://lmarena.ai/leaderboard

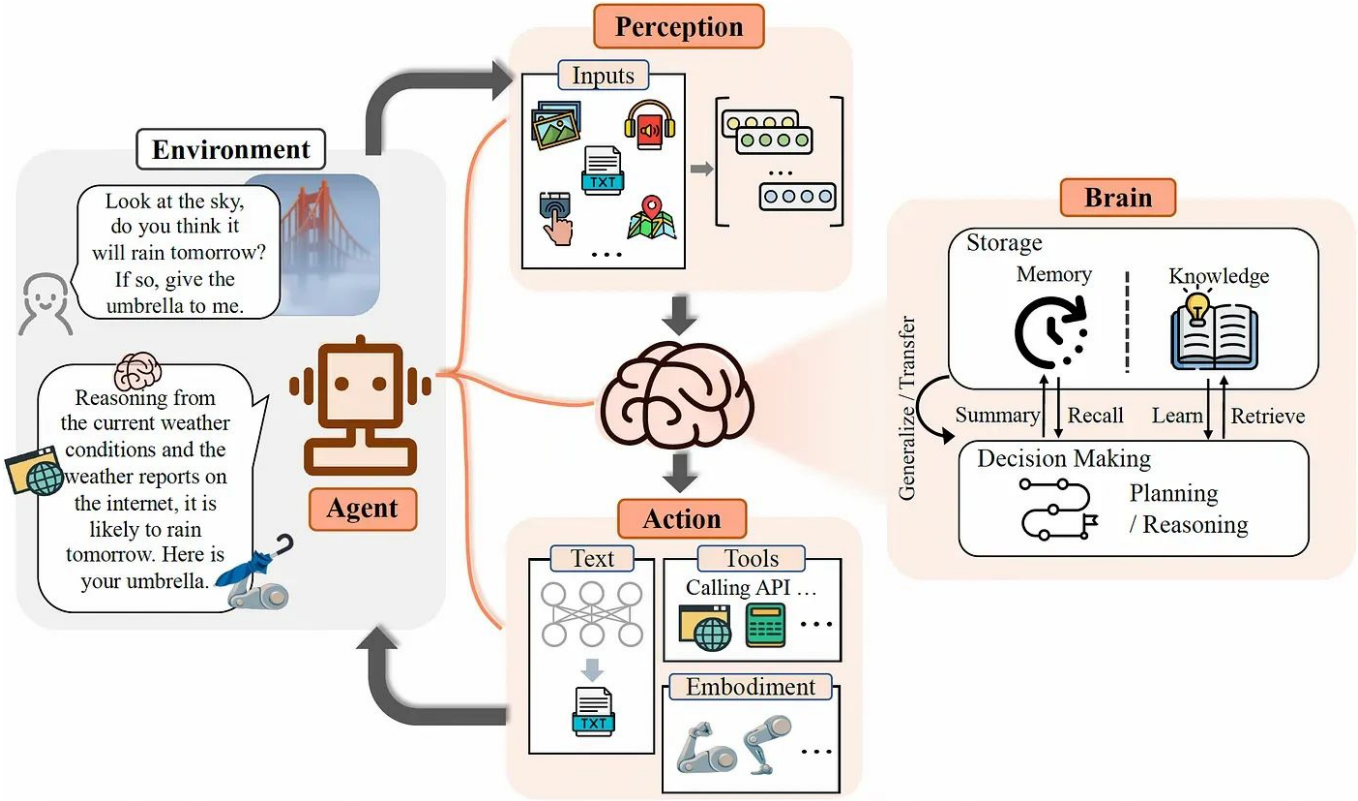


Model 276 / 276	Overall	Hard Prompts	Coding	Math	Creative Writing	Instruction Following	Longer Query	Multi-Turn
gemini-3-pro	1	1	1	1	1	1	1	1
grok-4.1-thinking	1	2	1	1	2	3	6	2
claude-opus-4-5-202…	2	1	1	1	2	1	1	1
claude-opus-4-5-202…	3	1	1	-	2	1	1	1
gpt-5.1-high	3	2	4	1	2	3	3	2
grok-4.1	3	2	4	4	3	10	8	2
claude-sonnet-4-5-2…	5	2	1	1	2	2	1	2
gemini-2.5-pro	5	7	12	1	2	4	6	3
claude-opus-4-1-202…	6	2	1	4	2	2	1	2
claude-sonnet-4-5-2…	6	4	2	4	2	3	3	1
claude-opus-4-1-202…	8	5	4	4	3	3	6	2
gpt-4.5-preview-202…	8	14	14	8	2	8	8	2
gpt-5.1	8	11	5	7	9	9	6	2

AI Agent vs. Agentic AI

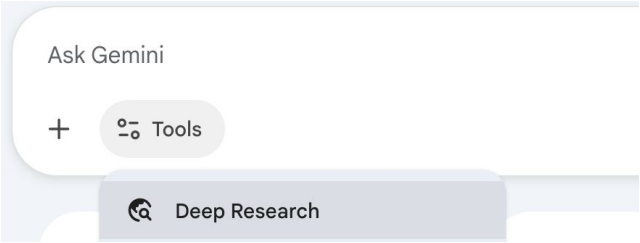
- AI Agents are broad, autonomous systems driven by goals.
- Agentic AI is a specific LLM pattern for task completion.
- Agentic AI uses planning, memory, and tool usage with an LLM.
- AI agents can be simple or utility-based, using many models.

AI Agents


Perception
Inputs
TXT
Environment
Look at the sky, do you think it will rain tomorrow? If so, give the umbrella to me.
Reasoning from the current weather conditions and the weather reports on the internet, it is likely to rain tomorrow. Here is your umbrella.
Agent
Brain
Storage
Memory
Knowledge
Generalize / Transfer
Summary
Recall
Learn
Retrieve
Decision Making
Planning
/ Reasoning
Action
Text
Tools
Calling API …
Embodiment
TXT

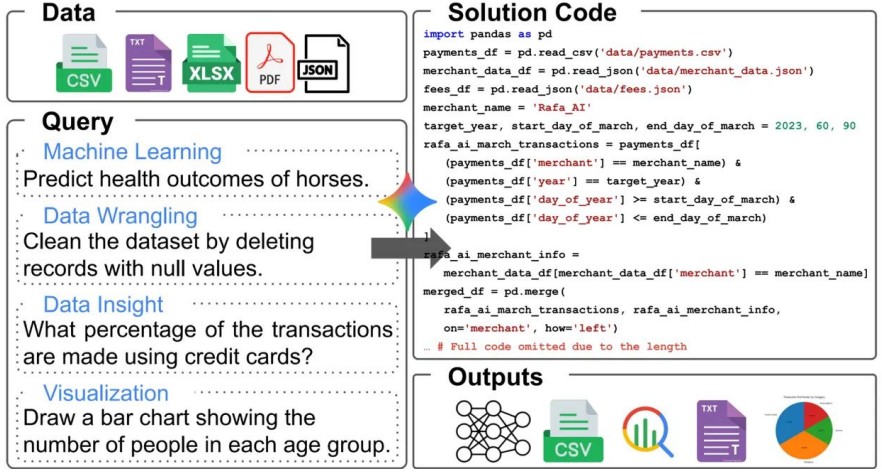
AI Agents

- Deep Research
  - Your personal research assistant



- Storybook Generation

- Data Science



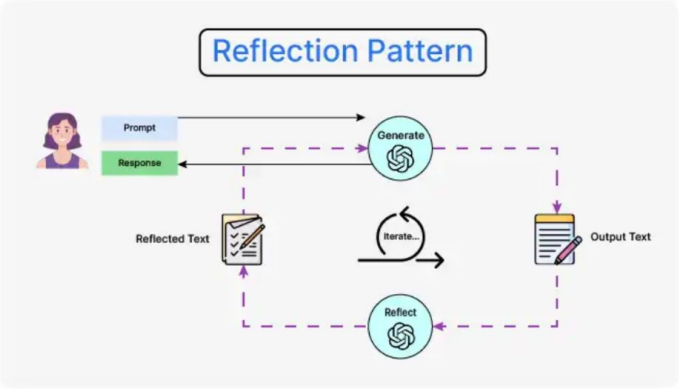
- Computer Use Agent

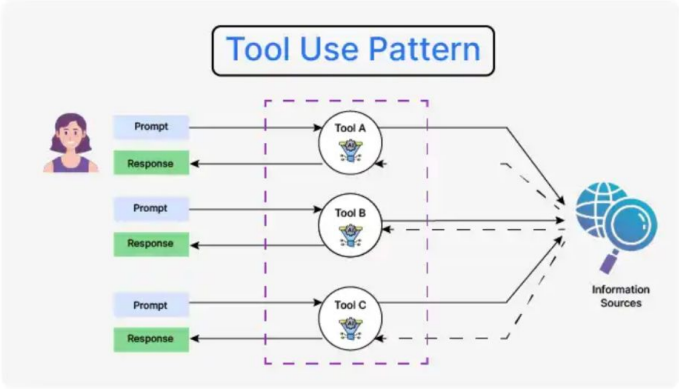
Agentic Reasoning Design Patterns

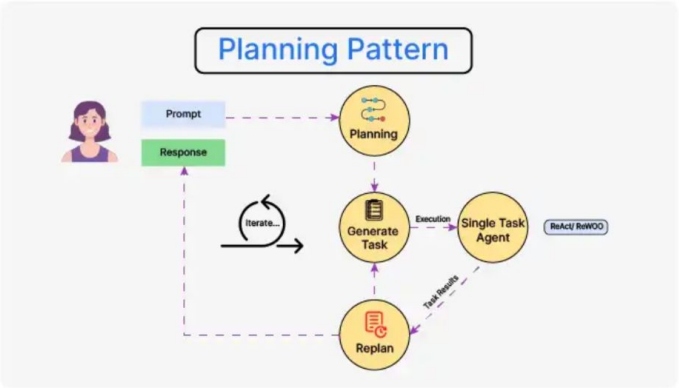
1. Reflection
   - Self-Refine: Iterative Refinement with Self-Feedback, Madaan et al. (2023)
   - Reflexion: Language Agents with Verbal Reinforcement Learning, Shinn et al., (2023)
2. Tool use
   - Gorilla: Large Language Model Connected with Massive APIs, Patil et al. (2023)
   - MM-REACT: Prompting ChatGPT for Multimodal Reasoning and Action, Yang et al. (2023)
3. Planning
   - Chain-of-Thought Prompting Elicits Reasoning in Large Language Models, Wei et al., (2022)
   - HuggingGPT: Solving AI Tasks with ChatGPT and its Friends in Hugging Face, Shen et al. (2023)
4. Multi-agent collaboration
   - Communicative Agents for Software Development, Qian et al., (2023)
   - AutoGen: Enabling Next-Gen LLM Applications via Multi-Agent Conversation, Wu et al. (2023)

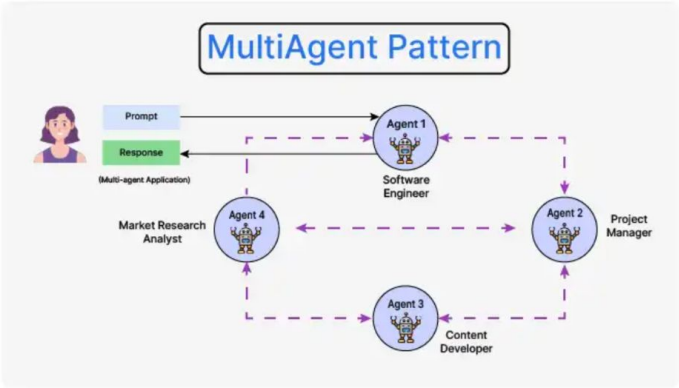
Andrew Ng

Agent Patterns





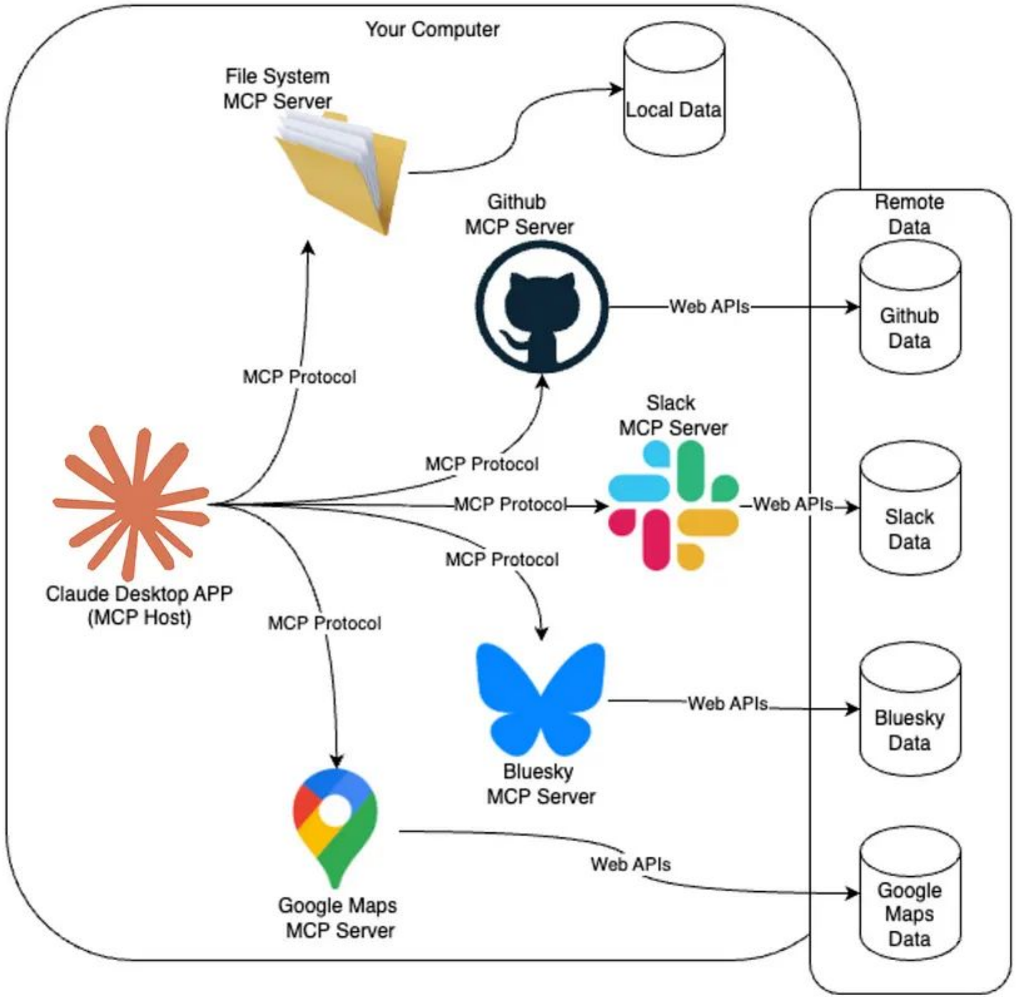




https://www.analyticsvidhya.com/blog/2024/10/agentic-design-patterns/

Agent Protocol

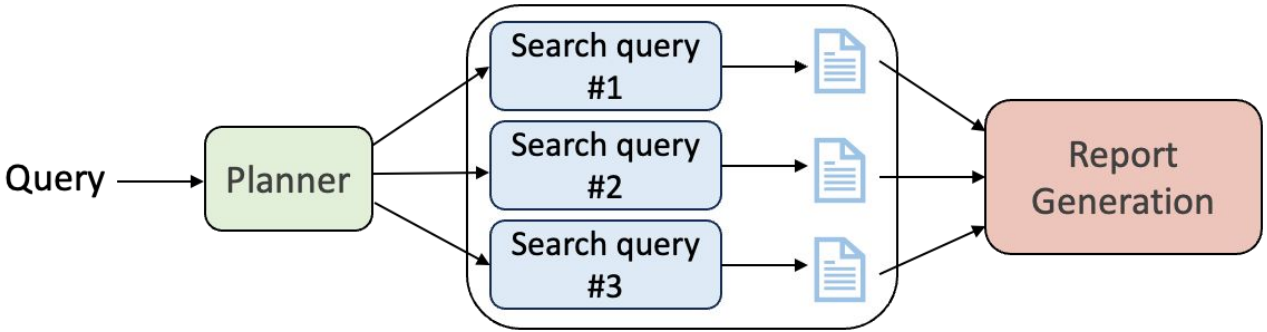
- MCP (Anthropic)
- A2A (Google)



Deep Research Agent

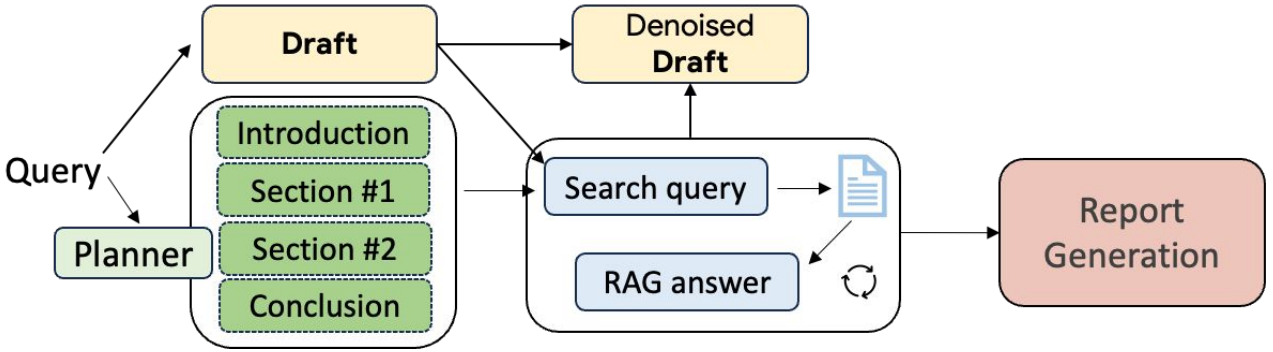
Huggingface Open DR

GPT Researcher



Deep Research Agent

Gemini Enterprise DR (ours)



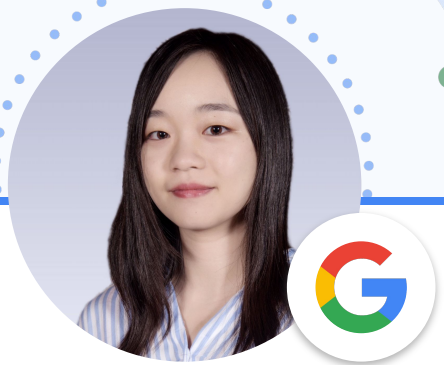
https://arxiv.org/abs/2507.16075

https://google.github.io/adk-docs/

Agent Framework

- Google ADK

Agent Development Kit



Yinghan Long

- PhD - Purdue, Bachelor's - SJTU and UMich
- MLE @Google Gemini Enterprise
- Research interests: AI agents, Deep learning, NLP, CV

Q&A